

Research Article

Reorganizing Neural Network System for Two Spirals and Linear Low-Density Polyethylene Copolymer Problems

G. M. Behery,¹ A. A. El-Harby,¹ and Mostafa Y. El-Bakry²

¹ *Mathematics Department, Faculty of Science, Mansoura University, New Damietta, Egypt*

² *Physics Department, Faculty of Science, Benha University, Al Qalyubiyah, Egypt*

Correspondence should be addressed to A. A. El-Harby, elharby@yahoo.co.uk

Received 28 February 2009; Revised 24 October 2009; Accepted 12 November 2009

Recommended by Zhigang Zeng

This paper presents an automatic system of neural networks (NNs) that has the ability to simulate and predict many of applied problems. The system architectures are automatically reorganized and the experimental process starts again, if the required performance is not reached. This processing is continued until the performance obtained. This system is first applied and tested on the two spiral problem; it shows that excellent generalization performance obtained by classifying all points of the two-spirals correctly. After that, it is applied and tested on the shear stress and the pressure drop problem across the short orifice die as a function of shear rate at different mean pressures for linear low-density polyethylene copolymer (LLDPE) at 190°C. The system shows a better agreement with an experimental data of the two cases: shear stress and pressure drop. The proposed system has been also designed to simulate other distributions not presented in the training set (predicted) and matched them effectively.

Copyright © 2009 G. M. Behery et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. Introduction

Neural networks are widely used for solving many problems in most science problems of linear and nonlinear cases [1–9]. Neural network algorithms are always iterative, designed to step by step minimise (targeted minimal error) the difference between the actual output vector of the network and the desired output vector, examples include the Backpropagation (BP) algorithm [10–12] and the Resilient Propagation (RPROP) algorithm [13–15].

Neural classifiers can deal with many multivariable nonlinear problems for which an accurate analytical solution is difficult to obtain. It is found however that the use of neural classifiers depends on several parameters that are crucial to the accurate predictions of the properties sought. The appropriate neural architecture, the number of hidden layers, and the number of neurons in each hidden layer are issues that can greatly affect the accuracy of the prediction. Unfortunately, there is no direct method to specify these factors as they need to be determined on experimental and trial basis [16].

The two-spiral benchmark was considered as one of the most difficult problems in two-class pattern classification field due to the complicated decision boundary [17]. It is

extremely hard to solve using multilayer perceptron models trained with various BP algorithms [18]. Thus, it is a well-known benchmark problem for testing the quality of neural network classifiers [19].

The effects of pressure on the viscosity and flow stability of one of the commercial grade polyethylenes (PEs) which is linear low-density polyethylene copolymer have been studied. The range of shear rates considered covers both stable and unstable flow regimes. “Enhanced exit-pressure” experiments have been performed attaining pressures of the order of 500×10^5 Pa at the die exit. The necessary experimental conditions have been clearly defined so that dissipative heating can be neglected.

Very high pressures can be exerted on polymers during processing. At these pressure levels, polymer melt properties, and flow stability, evolve according to laws that are different from those used at moderate pressures. In the following work by Couch and Binding [20], temperature and pressure dependence of shear stress can be modeled. Carreras et al. [21] studied these effects experimentally using different rheometers. The data obtained by Carreras et al. [21] is chosen to be carried out using the neural networks depending on the BP and RPROP algorithms.

BP is the most widely used algorithm for supervised learning with multilayered feed-forward networks [22], and it is very well known, while the RPROP algorithm is not well known and described in some detail in Section 3.1.

The RPROP algorithm was faster than the BP [23, 24]. Therefore, the RPROP is chosen to be carried out in this study. The present work offers an efficient neural network that is used to predict the unknown data of shear stress and the pressure drop across the short orifice die as a function of shear rate at different mean pressures for linear low-density polyethylene copolymer at 190°C. The following sections provide a brief introduction to NNs, describe the selected NN structure, training data, and discuss the results.

2. The Studied Problems

2.1. Two Spirals. The two-spiral problem is a classification task that consists of deciding in which of two interlocking spiral-shaped regions a given coordinate lies. The interlocking spiral shapes are chosen for this problem because they are not linearly separable. Finding a neural network solution to the two-spirals problem has proven to be very difficult when using a traditional gradient-descent learning method such as backpropagation, and therefore it has been used in a number of studies to test new learning methods; see for instance, [25, 26].

To learn and solve this task, a training set consists of 194 preclassified coordinates. Half of the coordinates is located in one spiral-shaped region and marked with triangles, and the other spiral-shaped region marked with circles. The coordinates of the 97 triangles are generated using the following equations, where $i = 0, 1, \dots, 96$. The coordinates of the circles are generated simply by negating the coordinates of the triangles [27]

$$r = \frac{6.5(104 - i)}{104}, \quad \theta = i \frac{\pi}{16}, \quad x = r \sin \theta, \quad y = r \cos \theta. \quad (1)$$

When performing a correct classification, the neural network takes two inputs corresponding to an (x, y) coordinate, and produces a positive signal if the point falls within the spiral that drawn using triangles and a negative signal if the point falls within the spiral that drawn by circles [27].

2.2. Linear Low-Density Polyethylene Copolymer. The studied problem consists of two dependent parts, the first is the pressure drop across the short orifice die as a function of shear rate at different mean pressures for linear low-density polyethylene copolymer at 190°C and the second is the shear stress dependence on shear strain of the same fluid at 190°C which represent flow curves. Each part contains seven groups of data. Each group has some samples as specified in [21]. The group data number 5 is specified to be predicted for each part, while the other six groups are chosen as patterns for training. The six groups for each part are prepared as input patterns of the proposed neural network algorithm.

This problem has two inputs (mean pressure and γ share rate), and single output (pressure drop Δp (Pa)) in each part,

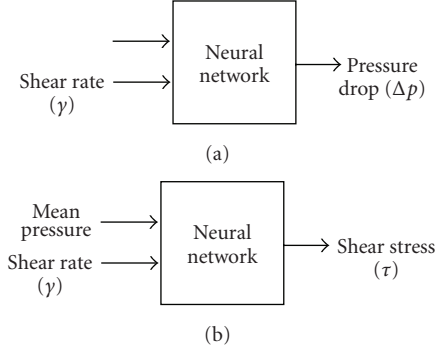


FIGURE 1: A block diagram modeling.

because there is only one target value associated with each input vector; see Figure 1.

3. Neural Networks

Neural networks consist of a number of units (neurons) which are connected by weighted links. These units are typically organised in several layers, namely, an input layer, one or more hidden layers, and an output layer. The input layer receives an external activation vector and passes it via weighted connections to the units in the first hidden layer. Figure 2 shows input layer with R elements, one hidden layer with S neurons, and output layer with one element. Each neuron in the network is a simple processing unit that computes its activation $y_i^{(1)}$ with respect to its incoming excitation, the so-called net input net_i ,

$$net_i = \sum_{j \in \text{pred}} s_j w_{ij} - b_i, \quad (2)$$

where $\text{pred}(i)$ denotes the set of predecessors of unit i , w_{ij} denotes the connection weight from unit j to i unit, and b_i is the unit bias value. The activation of unit i , $y_i^{(1)}$, is computed by passing the net input through a non-linear activation function. The tan-sigmoid function is applied in the proposed work as follows:

$$y_i^{(1)} = f_{\tan \text{ sig}}(net_i) = \frac{1}{1 + e^{-2net_i}} - 1. \quad (3)$$

3.1. RPROP Algorithm. In the RPROP algorithm, each weight (w_{ij}) is computed by its individual update-value ($\Delta_{ij}^{(t)}$), which determines the size of the weight update. This adaptive update-value evolves during the learning process based on its local sight on the error function E , according to the following learning-rule [13]:

$$\Delta_{ij}^{(t)} = \begin{cases} \eta+ * \Delta_{ij}^{(t-1)} & \text{if } \frac{\partial E^{(t-1)}}{\partial w_{ij}} * \frac{\partial E^{(t)}}{\partial w_{ij}} > 0, \\ \eta- * \Delta_{ij}^{(t-1)} & \text{if } \frac{\partial E^{(t-1)}}{\partial w_{ij}} * \frac{\partial E^{(t)}}{\partial w_{ij}} < 0, \\ \Delta_{ij}^{(t-1)}, & \text{else,} \end{cases} \quad (4)$$

where $0 < \eta- < 1 < \eta+$.

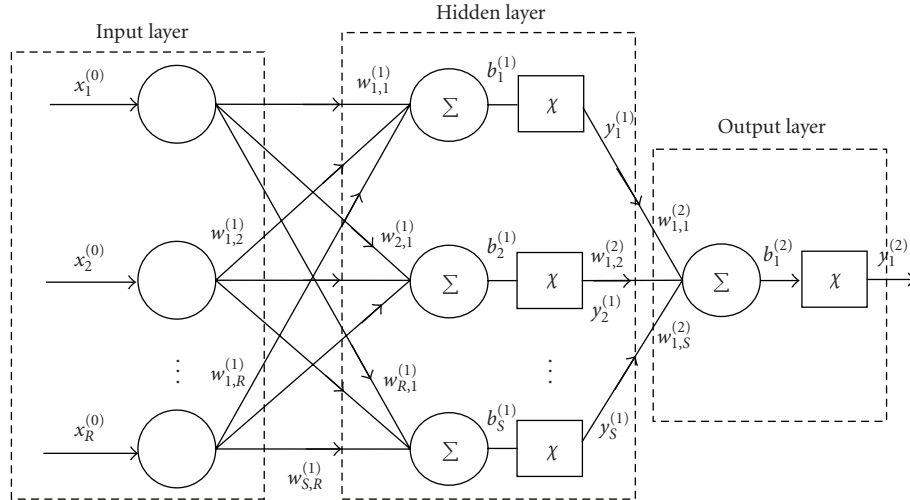


FIGURE 2: Network Architecture for one hidden layer.

The size of the weight change is exclusively determined by the weight-specific update-value $\Delta_{ij}^{(t)}$. Every time the partial derivative of the corresponding weight changes its sign, the update-value is decreased by the factor η . This indicates that the last update was too big and the algorithm jumped over a local minimum. On the other hand, if the derivative retains its sign the update-value is slightly increased by the factor η in order to accelerate convergence in shallow regions. Once the update-value for each weight is adapted, the weight-update is changed as follows: if the derivative is positive (increasing error) the weight is decreased by its update-value, if the derivative is negative, the update-value is added. Then, the weights are updated as in (6) using update-values from

$$\Delta_{ij}^{(t)} = \begin{cases} -\Delta_{ij}^{(t)} & \text{if } \frac{\partial E^{(t)}}{\partial w_{ij}} > 0, \\ +\Delta_{ij}^{(t)} & \text{if } \frac{\partial E^{(t)}}{\partial w_{ij}} < 0, \\ 0, & \text{else,} \end{cases} \quad (5)$$

$$w_{ij}^{t+1} = w_{ij}^{(t)} + \Delta w_{ij}^{(t)} \quad (6)$$

see [23].

As mentioned in the end of Section 1, the RPROP algorithm was faster than the BP, the main reason for the success using this algorithm is that the size of weight-step is only dependent on the sequence of signs, not on the magnitude of the derivative as showed by Riedmiller and Braun [23]. The RPROP algorithm has fewer parameters that need to be evaluated and promises to provide the same performance as an optimally trained network using the BP algorithm.

3.2. Proposed System. The proposed system is designed to work in automatic way starting with random initial weighed and biases values. Many NN experiments are done to have the optimal NN results of the two-spiral problem, by repeating the same experiment using the same NN architecture

(number of hidden layers and neurons). Therefore, 500 NN experiments and 400 neurons are specified as maximum numbers of this system. The system stops when the best network is obtained. This system is trained and tested using different parameters, for instance, changing the number of hidden layers, neurons, and epochs. The experimental data of the two physical problems (the shear stress and the pressure drop problem across the short orifice die as a function of shear rate at different mean pressures for linear low-density polyethylene copolymer (LLDPE) at 190°C) are smaller. Therefore, the optimal required numbers of hidden layers and neurons are specified by NN experiments depending on trial and error as applied, for instance, in [28, 29]. If the required performance is not reached in the test process, this system is continued to do another new experiment. When the last NN experiment is reached, the number of neurons for each hidden layer is incremented in sequence way and the process of another 500 NN experiments starts again. The incremented process is continued until the required performance is reached. If not, and the maximum number of neurons is reached, an alternative way is started; the number of hidden layers is incremented by one and new 500 NN experiments start again with initializing the number of neurons for these hidden layers. The system is continuous until excellent training and prediction is reached. The details of the proposed system are shown in Figure 3.

This proposed system is based on RPROP algorithm using tan-sigmoid transfer function in the hidden layers and a linear transfer function in the output layer. More hidden layers or neurons require more computations, but allow the network to solve complicated problems. Therefore, many tries are done to find the best network that uses low number of hidden layers and low number of neurons.

After the training, in the test process of the two-spirals problem, it is noticed that the chosen algorithm using two hidden layers with 77 neurons for each one is very effective for reaching the optimal classification; see Figure 4(a). While in the test process of the both pressure drop and shear

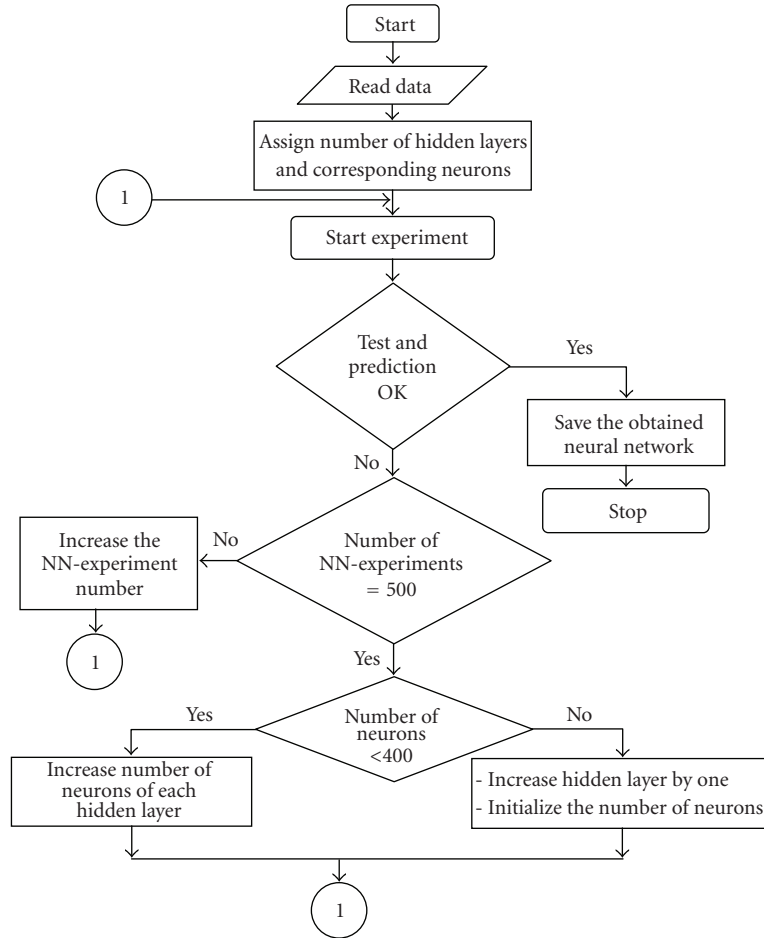


FIGURE 3: The proposed system diagram.

stress problems, it is found that one-hidden layer having 20 neurons is enough for reaching the optimal performance as specified in Figure 4(b). We first set up in Figure 4 the network in the proposed system with random weights and biases values, where $IW\{1, 1\}$ represents the input weights, $LW\{2, 1\}$ and $LW\{3, 2\}$ mean the layer weights, $b\{1\}$ is the biases of the input layer, and $b\{2\}$ and $b\{3\}$ are the biases of the output layers. The obtained weights and biases of the best trained network for pressure drop and shear stress problems are shown in Table 2.

4. Results

The proposed system is carried out on three problems. They are two-spiral problem, the pressure drop and shear stress across the short orifice die as a function of shear rate at different mean pressures for linear low-density polyethylene copolymer at 190°C . The descriptions of the obtained results are showed in the following three subsections.

4.1. Two-Spiral Problem. This problem is used to learn a mapping function (two inputs and one output) which distinguishes points on two intertwined spirals. This is one of

TABLE 1: The NN evaluations.

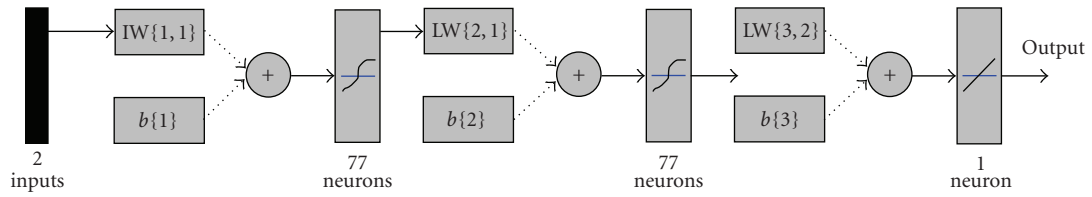
Experiments	▲	●	Accuracy
1-HL & 400 neurons	6	8	92.8%
2-HL & 50 neurons	6	2	95.9%
2-HL & 60 neurons	1	0	99.5%
2-HL & 77 neurons	0	0	100%

▲ means no. misclassified of the triangles spiral.

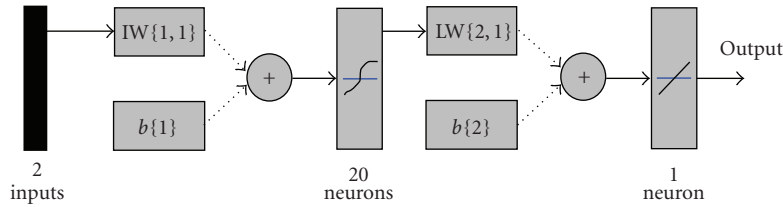
● means no. misclassified of the circles spiral.

the typical difficult problems due to its extreme nonlinearity. The proposed system was first trained on 194 points of the X - Y coordinates using one hidden layer. The obtained performance was 92.8% at 2157 epochs and 400 neurons; see Figure 6(a) and Table 1. The misclassified points are 6 in the triangles spiral and 8 in the circles-spiral.

The training process is continued with increasing one hidden layer more. The obtained performances are 95.9%, 99.5%, and 100% using 50, 60, and 77 neurons, respectively; see Figures 6(b)–6(d) and Table 1. The numbers of training epochs for these obtained performances are 10000, 10000, and 5896, respectively (see Figure 5). It is found that all



(a) Two-Spirals problem.



(b) linear low-density polyethylene copolymer problem

FIGURE 4: The architecture of the proposed network.

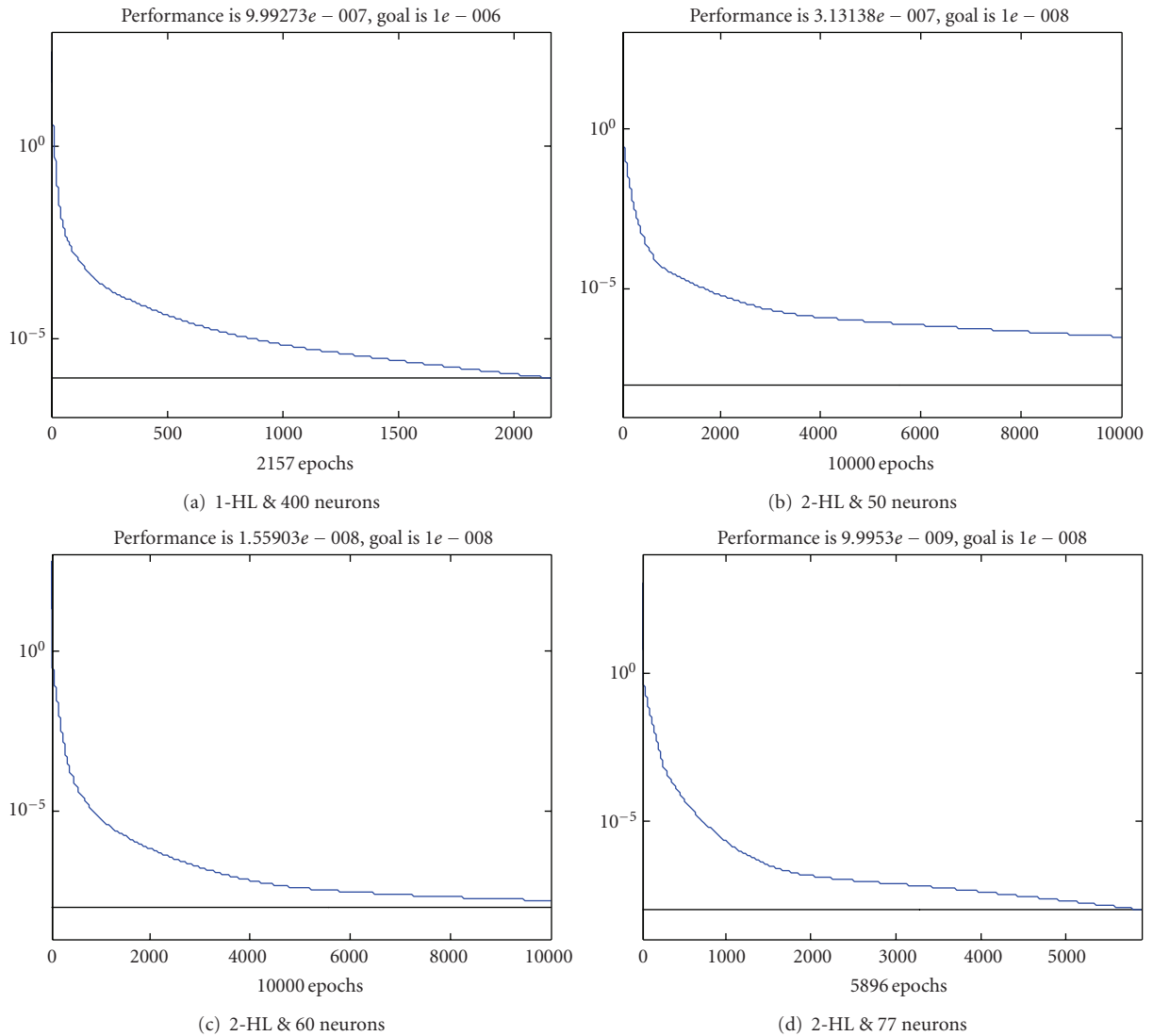


FIGURE 5: The NN performance (HL means hidden layer).

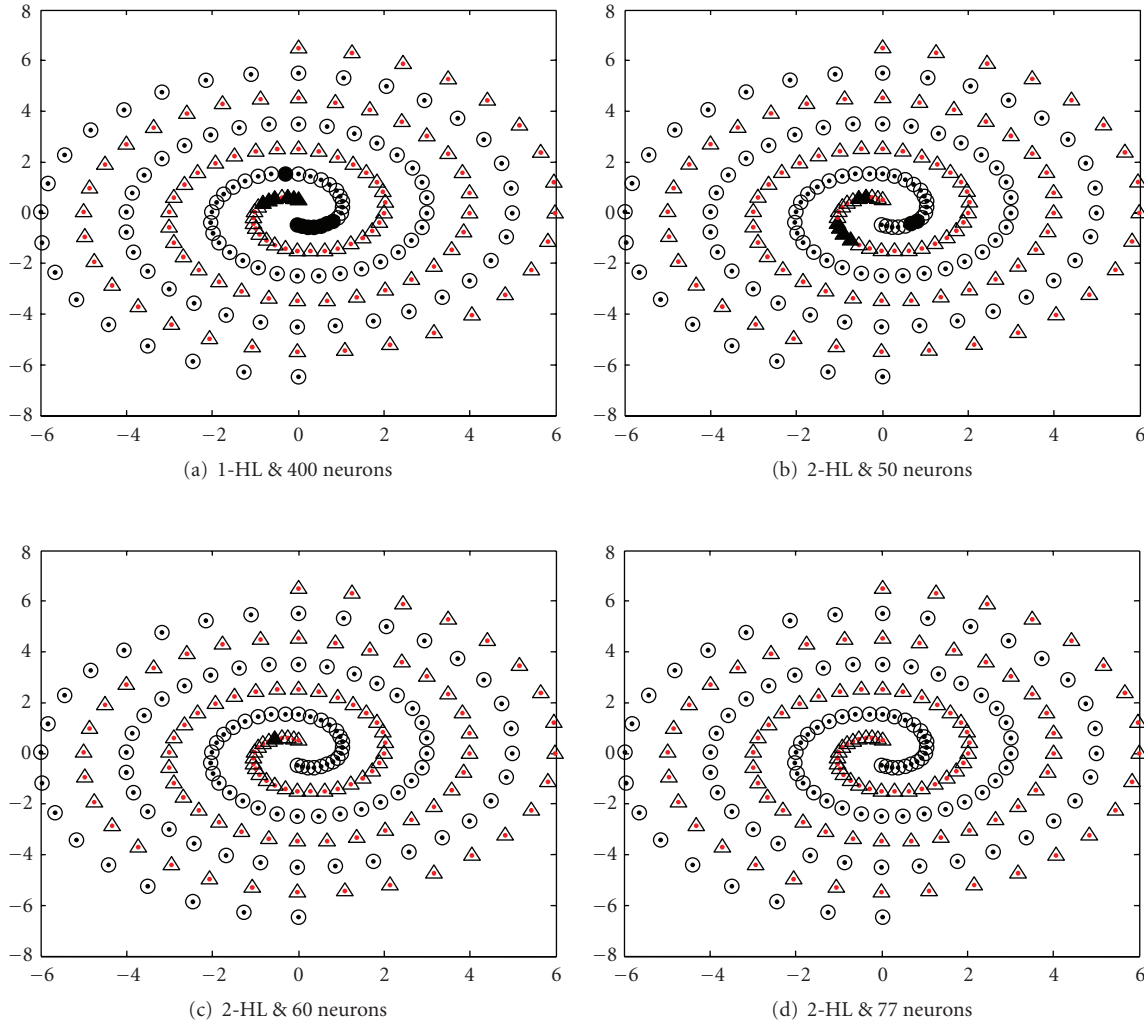


FIGURE 6: The NN simulation of the pressure drop dependence on shear rate.

194 training patterns are classified correctly using the last network architecture.

4.2. Linear Low-Density Polyethylene Copolymer. The above mentioned details of the proposed system were applied and simulated to the data of the pressure drop and shear stress across the short orifice die as a function of shear rate at different mean pressures for linear low-density polyethylene copolymer at 190°C , with one hidden layer.

The system was trained using the chosen neural network on six cases of different mean pressures for each of the pressure drop and shear stress as a function of shear rate. These values of mean pressure are 1, 100, 200, 300, 500, and 600 multiplied by 10^5 Pa. The performances of the obtained networks are shown in Figure 7. The obtained networks were tested for choosing the best one. This network was tested on the above mentioned six cases and used for predicting the case at mean pressure value, 400×10^5 Pa. Figure 8 shows the neural networks results of the six cases training and one

predicted case for pressure drop which denoted by Δp with shear rate. Figure 9 shows also the neural networks results of the six cases training and one predicted case for shear stress which denoted by τ with shear rate which represent flow curve. It was observed that these figures illustrate an excellent performance in two cases (the training and prediction). These results of the dependence of pressure drop and shear stress on shear rate at different mean pressures are presented in the following two subsections.

4.2.1. Pressure Dependence on Shear Rate. Figure 8 shows the six cases tested and one predicted data of the pressure drop with shear rate compared to the experiential data for linear low-density polyethylene copolymer at 190°C .

4.2.2. Shear Stress Dependence on Shear Rate (Flow Curve). Figure 9 shows the six cases tested and one predicted data of the shear stress with shear rate compared to the

TABLE 2: The produced weight and bias values of the best trained network.

Pressure drop					Shear sftress				
Weights			Biases		Weights			Biases	
IW{1, 1}	LW{2, 1} ^T		b {1}	b {2}	IW{1, 1}	LW{2, 1} ^T	b {1}	b {2}	
-0.0092357	-0.074211	-0.91753	16.5516	1.0932	0.42093	0.11045	-0.29861	-14.8788	0.49395
-0.064121	-0.075615	-1.238	-14.7538		0.14907	-0.028838	0.14388	-12.3761	
0.0086149	-0.025249	-2.0174	-2.5568		0.00031069	0.026445	-0.52767	-11.8835	
-0.0017662	0.036417	1.4997	-2.6596		-0.0013544	0.0019265	-0.71031	-1.7975	
0.023687	-0.0099672	0.58772	-3.0918		-2.385	-2.4141	0.38584	-7.6139	
-0.017128	0.0023562	-0.97796	7.6266		0.01941	0.0086871	0.089209	-11.4316	
-0.063098	-0.083557	-1.6479	-8.8354		0.0035304	-0.075417	-0.67559	-1.0939	
0.20564	0.021342	0.91681	-6.5694		-0.0076158	-0.028539	-0.19949	1.5115	
-0.003303	0.022274	1.7385	0.081723		0.044023	0.038829	0.017871	-10.0832	
-0.041185	0.079807	1.8449	-11.2574		0.005363	-0.024715	-0.49156	6.6207	
0.024424	-0.075788	-1.1288	-2.3592		-0.0013665	0.032209	0.18869	7.0804	
0.17343	0.14075	1.9772	1.7446		-0.00048342	-0.00098168	-2.8782	0.41403	
-0.011373	-0.02235	-2.4026	9.3959		0.00076532	-0.040128	-0.65477	-1.5022	
0.0038815	0.082769	0.97819	8.3238		-0.0081859	0.27396	0.15231	-4.8667	
0.0045642	0.081391	0.95285	7.9681		-0.012168	-0.012549	-0.14515	6.2695	
-0.0023447	-0.011416	-3.3245	6.9351		0.0038695	0.010562	-0.29916	-3.3411	
-0.17721	-0.15542	-1.3838	1.5062		-0.054214	-0.0096744	-0.19168	4.0609	
0.013042	0.020739	0.42706	-13.5254		-0.014118	0.015177	0.30145	-5.2238	
0.094386	0.060478	1.6078	11.2445		-0.0014922	0.024903	0.32635	-1.9256	
0.10412	0.10144	0.4335	5.2448		0.0046948	-0.030181	-0.32884	-0.096699	

LW{2, 1}^T represents the transpose of LW{2, 1}. All names in the above table were described in architecture of the proposed network; see Figure 4.

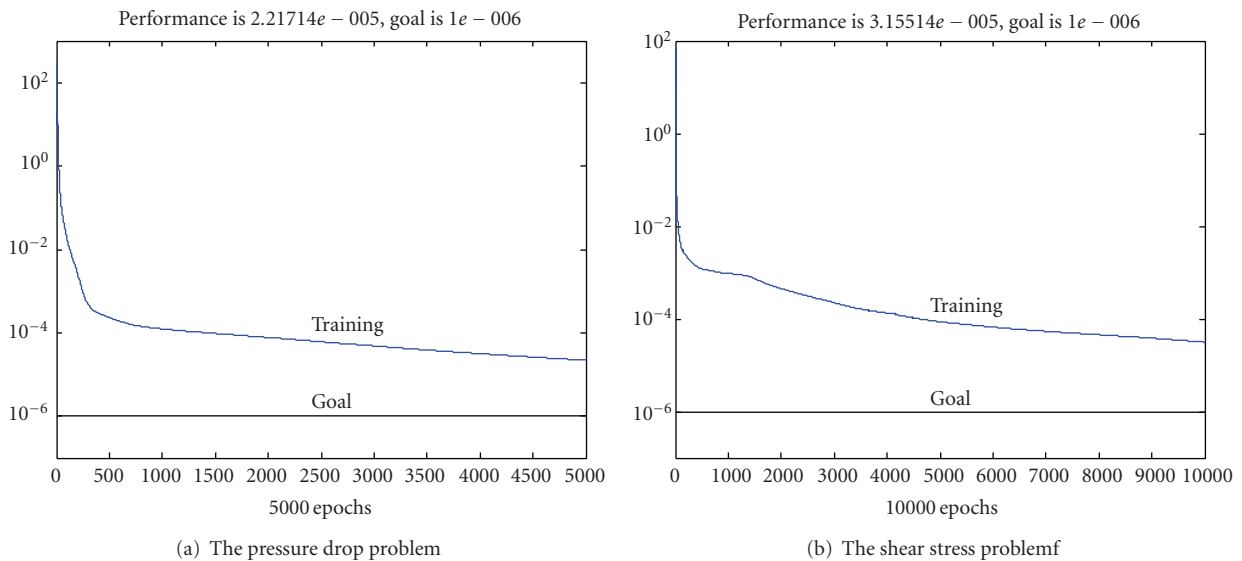


FIGURE 7: The NN performance.

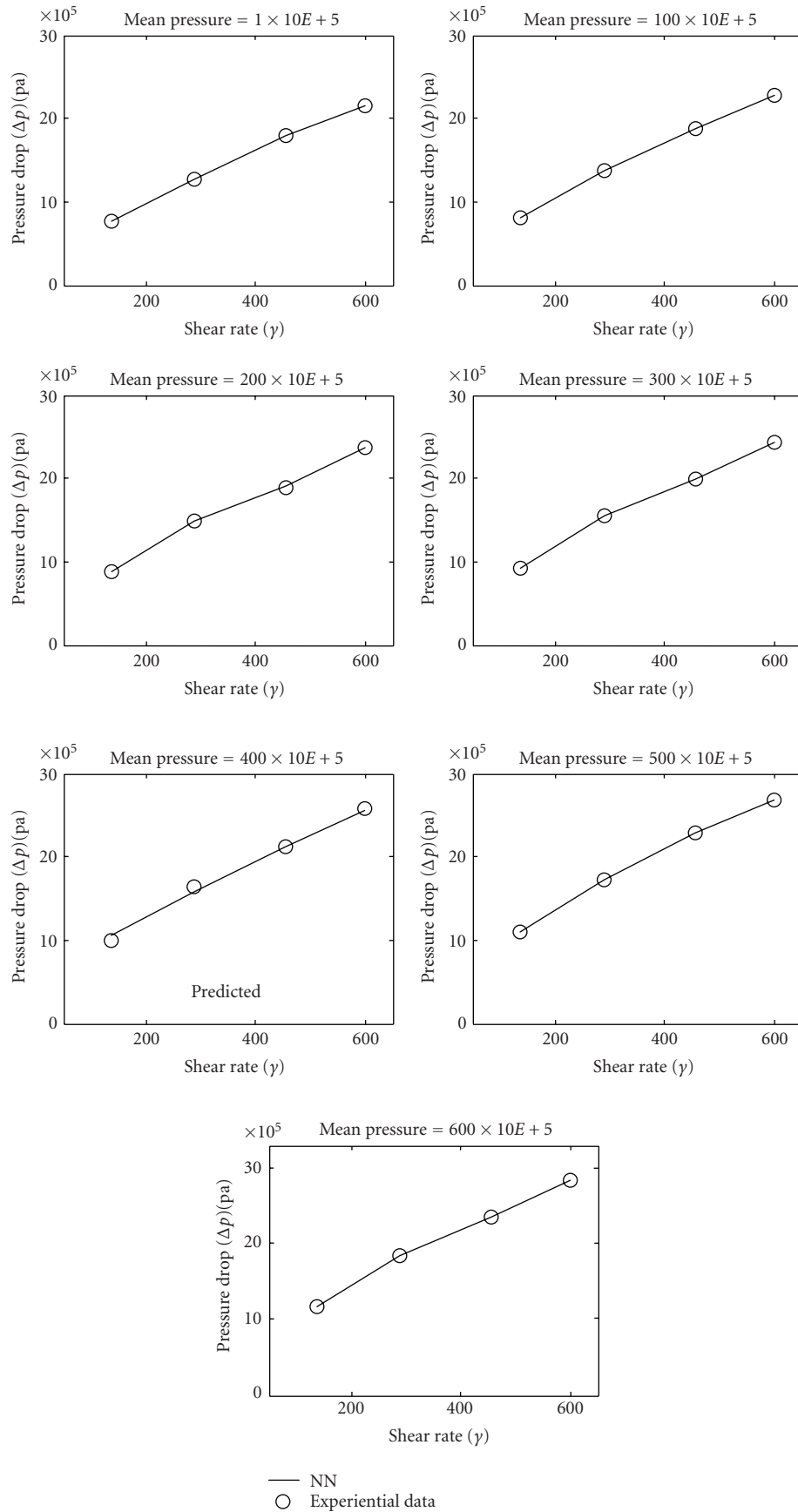


FIGURE 8: The NN simulation of the pressure drop dependence on shear rate.

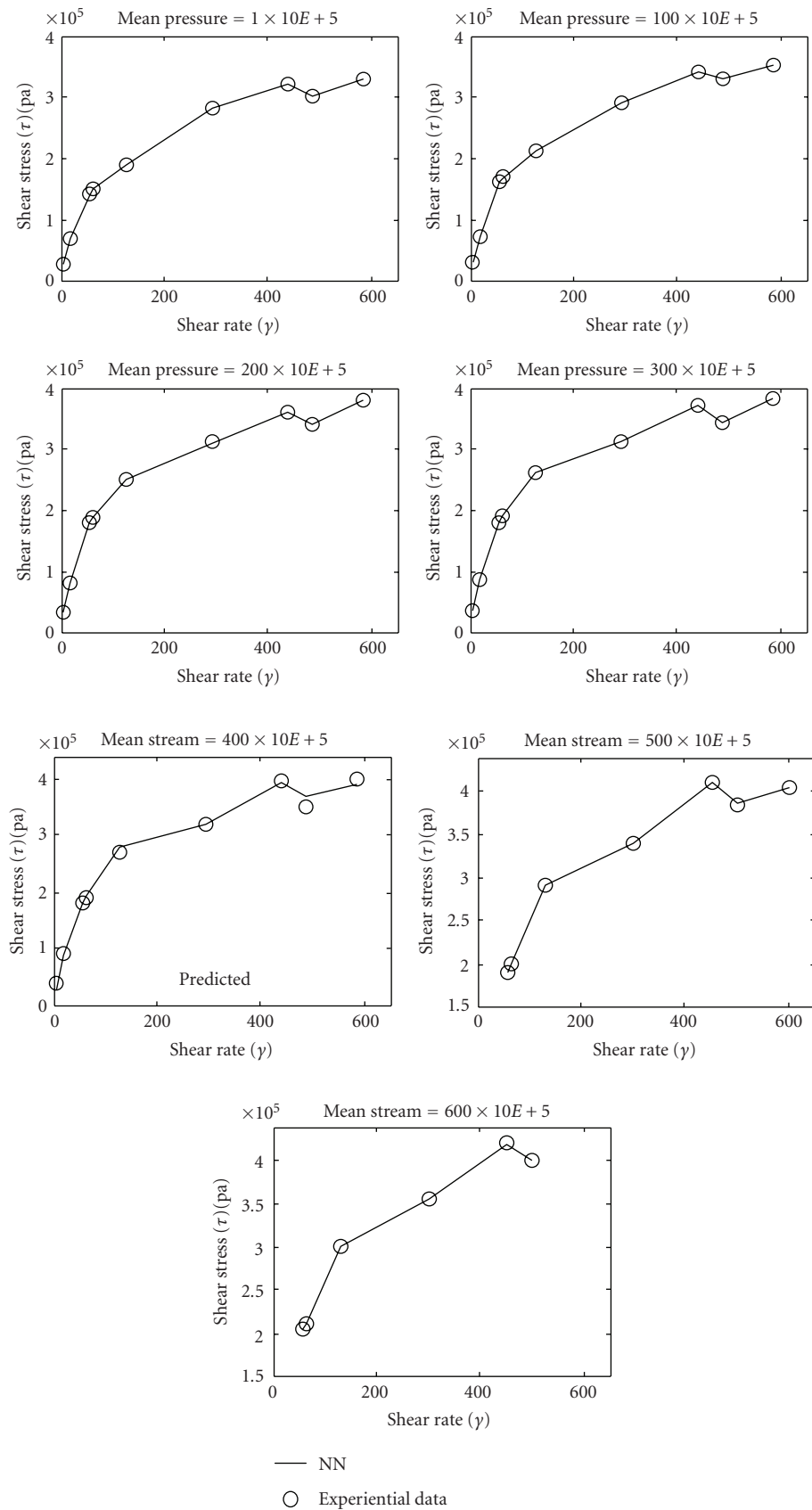


FIGURE 9: The NN simulation of the share stress dependence on shear rate.

experimental data which represent flow curve for linear low-density polyethylene copolymer at 190°C.

5. Conclusion

The proposed system is automatically designed to find the best network that has the ability to have the best test and prediction. This technique is started by doing 500 NN experiments with incrementing the number of neurons for each hidden layer. In the incrementing process, another new 500 NN-experiments are carried out in alternative way; the number of hidden layers is incremented by one with initializing the number of neurons for these hidden layers and new 500 NN-experiments are done. This process is continued until the required performance is reached. Therefore, many tries are automatically done to find this network, using low number of hidden layers and neurons.

The obtained performance of the two-spiral problem is low when using one hidden layer in the network architecture, although the number of neurons increased up to 400. The performance is improved using two hidden layers, it is 95.9% with 50 neurons, 99.5% with 60 neurons, and 100% with 77 neurons. In the best performance, all points of the two-spiral problem are correctly classified.

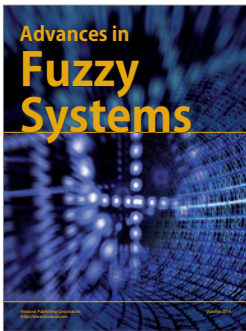
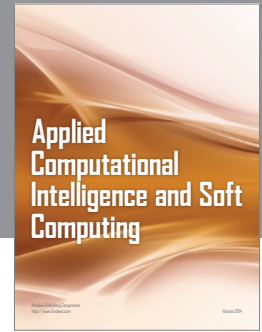
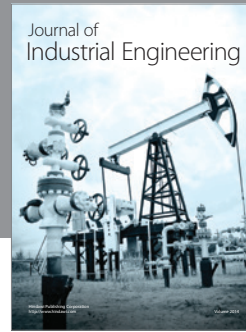
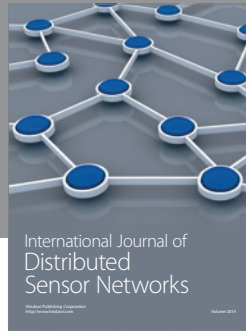
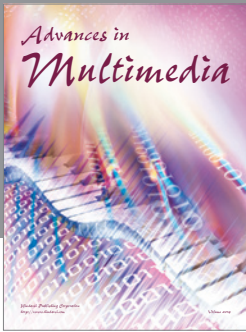
In the other two problems, it was found that one hidden layer with 20 neurons is enough for reaching the optimal solution. The trained NN using this system shows excellent results matched with the experimental data in the two cases of shear stress and pressure drop problems. The NN technique has been also designed to simulate the other distributions not presented in the training set and matched them effectively.

The NNs simulation using RPROP algorithm is powerful mechanism for classifying all points of the two spirals, and for the prediction flow curves (dependence of shear stress on shear rate) and pressure drop dependence of shear rate at a certain value of mean pressure across short orifice die for linear low-density polyethylene copolymer at 190°C.

References

- [1] K. Sreenivasa-Rao and B. Yegnanarayana, "Modeling durations of syllables using neural networks," *Computer Speech and Language*, vol. 21, pp. 282–295, 2007.
- [2] H. Altun, A. Bilgil, and B. C. Fidan, "Treatment of multi-dimensional data to enhance neural network estimators in regression problems," *Expert Systems with Applications*, vol. 32, no. 2, pp. 599–605, 2007.
- [3] Á. Silva, P. Cortez, M. F. Santos, L. Gomes, and J. Neves, "Mortality assessment in intensive care units via adverse events using artificial neural networks," *Artificial Intelligence in Medicine*, vol. 36, no. 3, pp. 223–234, 2006.
- [4] G. Manduchi, S. Marinetti, P. Bison, and E. Grinzato, "Application of neural network computing to thermal non-destructive evaluation," *Neural Computing & Applications*, vol. 6, no. 3, pp. 148–157, 1997.
- [5] M. Y. El-Bakry, A. A. El-Harby, and G. M. Behery, "Automatic neural network system for vorticity of square cylinders with different corner radii," *Journal of Applied Mathematics and Informatics*, vol. 26, no. 5–6, pp. 911–923, 2008.
- [6] M. Y. El-Bakry and K. A. El-Metwally, "Neural network model for proton-proton collision at high energy," *Chaos, Solitons and Fractals*, vol. 16, no. 2, pp. 279–285, 2003.
- [7] M. Y. El-Bakry, "Feed forward neural networks modeling for K-P interactions," *Chaos, Solitons and Fractals*, vol. 18, no. 5, pp. 995–1000, 2003.
- [8] A.-K. Hamid, "Scattering from a spherical shell with a circular aperture using a neural network approach," *Canadian Journal of Physics*, vol. 76, no. 1, pp. 63–67, 1998.
- [9] G. Scalabrin, C. Corbetti, and G. Cristofoli, "A viscosity equation of state for R123 in the form of a multilayer feedforward neural network," *International Journal of Thermophysics*, vol. 22, no. 5, pp. 1383–1395, 2001.
- [10] Y.-C. Hu and J.-F. Tsai, "Backpropagation multi-layer perceptron for incomplete pairwise comparison matrices in analytic hierarchy process," *Applied Mathematics and Computation*, vol. 180, no. 1, pp. 53–62, 2006.
- [11] B. Curry and P. H. Morgan, "Model selection in neural networks: some difficulties," *European Journal of Operational Research*, vol. 170, no. 2, pp. 567–577, 2006.
- [12] J. J. Steil, "Online stability of backpropagation-decorrelation recurrent learning," *Neurocomputing*, vol. 69, no. 7–9, pp. 642–650, 2006.
- [13] M. Riedmiller, "Advanced supervised learning in multi-layer perceptrons—from backpropagation to adaptive learning algorithms," *Computer Standards & Interfaces*, vol. 16, no. 3, pp. 265–278, 1994.
- [14] C. Igel and M. Hüsken, "Empirical evaluation of the improved Rprop learning algorithms," *Neurocomputing*, vol. 50, pp. 105–123, 2003.
- [15] A. A. El-Harby, "Automatic classification system of fires and smokes from the Delta area in Egypt using neural networks," *International Journal of Intelligent Computing and Information Science*, vol. 8, no. 1, pp. 59–68, 2008.
- [16] Y. Al-Assaf and H. El Kadi, "Fatigue life prediction of composite materials using polynomial classifiers and recurrent neural networks," *Composite Structures*, vol. 77, no. 4, pp. 561–569, 2007.
- [17] G. Horváth, "Kernel CMAC: an efficient neural network for classification and regression," *Acta Polytechnica Hungarica*, vol. 3, no. 1, pp. 5–20, 2006.
- [18] Y. C. Liang, D. P. Feng, H. P. Lee, S. P. Lim, and K. H. Lee, "Successive approximation training algorithm for feedforward neural networks," *Neurocomputing*, vol. 42, pp. 311–322, 2002.
- [19] Y. Quan and J. Yang, "Geodesic distance for support vector machines," *Acta Automatica Sinica*, vol. 31, no. 2, pp. 202–208, 2005.
- [20] M. A. Couch and D. M. Binding, "High pressure capillary rheometry of polymeric fluids," *Polymer*, vol. 41, no. 16, pp. 6323–6334, 2000.
- [21] E. S. Carreras, N. El Kissi, J.-M. Piau, F. Toussaint, and S. Nigen, "Pressure effects on viscosity and flow stability of polyethylene melts during extrusion," *Rheologica Acta*, vol. 45, no. 3, pp. 209–222, 2006.
- [22] P. Frasconi, M. Gori, and G. Soda, "Links between LVQ and backpropagation," *Pattern Recognition Letters*, vol. 18, no. 4, pp. 303–310, 1997.
- [23] M. Riedmiller and H. Braun, "A direct adaptive method for faster backpropagation learning: the RPROP algorithm," in *Proceedings of IEEE International Conference on Neural Networks (IJCNN '93)*, H. Ruspini, Ed., pp. 586–591, San Francisco, Calif, USA, April 1993.

- [24] A. A. El-Harby, *Automatic extraction of vector representations of line features from remotely sensed images*, Ph.D. thesis, Keele University, Keele, UK, 2001.
- [25] G. A. Papakostas, Y. Boutalis, and S. Samartzidis, "Two-stage hybrid tuning algorithm for training neural networks in image vision applications," *The International Journal of Signal and Imaging Systems Engineering*, vol. 1, no. 1, pp. 58–67, 2008.
- [26] D. Tian, Y. Liu, and J. Wang, "Fuzzy neural network structure identification based on soft competitive learning," *International Journal of Hybrid Intelligent Systems*, vol. 4, no. 4, pp. 231–242, 2007.
- [27] M. A. Potter and K. A. De Jong, "Cooperative coevolution: an architecture for evolving coadapted subcomponents," *Evolutionary Computation*, vol. 8, no. 1, pp. 1–29, 2000.
- [28] K. J. Cios, W. Pedrycz, R. W. Swiniarski, and L. A. Kurgan, *Data Mining: A Knowledge Discovery Approach*, Springer Science+Business Media, LLC, New York, NY, USA, 2007.
- [29] J. Plaza, A. Plaza, R. Perez, and P. Martinez, "On the use of small training sets for neural network-based characterization of mixed pixels in remotely sensed hyperspectral images," *Pattern Recognition*, vol. 42, no. 11, pp. 3032–3045, 2009.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

